RECEIVED

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/338,035	06/22/1999	HENRY ESMOND BUTTERWORTH	UK999026	9863

7590 09/11/2003

JAY P SBROLLINI
IBM CORP IP LAW DEPT
T J WATSON RESEARCH CENTER
P O BOX 218
YORKTOWN HEIGHTS, NY 10598

EXAMINER	
TANG, KENNETH	
ART UNIT	**PAPER NUMBER**
2127	12

DATE MAILED: 09/11/2003

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	09/338,035	BUTTERWORTH, HENRY ESMOND	
	Examiner	Art Unit	
	Kenneth Tang	2127	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
- Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *15 August 2003* .

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-18* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-18* is/are rejected.

7)☒ Claim(s) *6 and 7* is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☒ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

11)☐ The proposed drawing correction filed on _____ is: a)☐ approved b)☐ disapproved by the Examiner.

If approved, corrected drawings are required in reply to this Office action.

12)☐ The oath or declaration is objected to by the Examiner.

Priority under 35 U.S.C. §§ 119 and 120

13)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____ .

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

14)☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. § 119(e) (to a provisional application).

a) ☐ The translation of the foreign language provisional application has been received.

15)☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. §§ 120 and/or 121.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449) Paper No(s) _____ .
4) ☐ Interview Summary (PTO-413) Paper No(s). _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: .

DETAILED ACTION

1. This non-final action is in response to paper number 11, "Amendment A," which was received on 8/15/03. Claims 1-18 are presented for examination.

Drawings

2. The drawings are objected to under 37 CFR 1.83(a). The drawings must show every feature of the invention specified in the claims. Therefore, the "a computer program product embodied in a tangible media", "if the task code is not capable of fully loading into the instruction cache, logically dividing the task code such that at least one substantially atomic portion of the task code will fully load in the instruction cache" and "executing the task code for processing the new task in the instruction cache without loading new code into the instruction cache" must be shown or the feature(s) canceled from the claim(s). No new matter should be entered.

A proposed drawing correction or corrected drawings are required in reply to the Office action to avoid abandonment of the application. The objection to the drawings will not be held in abeyance.

Claim Objections

3. Claims 6 and 7 are objected to under 37 CFR 1.75(c) as being in improper form because a multiple dependent claim *should refer to other claims in the alternative only*. See MPEP § 608.01(n). Accordingly, the claims 6 and 7 have not been further treated on the merits.

Claim Rejections - 35 USC § 112

The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

4. The term "substantially" in claims 11-12, 14-16, and 18 is a relative term which renders the claim indefinite. The term "substantially" is not defined by the claim, the specification does not provide a standard for ascertaining the requisite degree, and one of ordinary skill in the art would not be reasonably apprised of the scope of the invention. And specifically in claims 14 and 18, "substantially atomic" are two contradictory terms. "Substantially" refers to an ample amount, while "atomic" refers to all or nothing.

5. Claims 6 and 7 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

6. Claims 1, 3, and 8-9 are rejected under 35 U.S.C. 103(a) as being unpatentable over Gulsen (US 5,727,211) in view of Kirk (US 5,875,464), and further in view of Klein (US 5,835,763).

7. Referring to claims 1 and 8-9, Gulsen discloses a method and system for processing tasks in a data processing system *("data processing system", col 3, lines 23-34)* including:

- an instruction cache *("instruction cache", col 3, lines 23-27)*;
- each task type having code associated with it *("one code segment per task", col 7, lines 44-45)*;

Gulsen fails to explicitly teach:

- tasks of different types are defined in the system

However, Kirk teaches executing tasks with different legal partition sizes. It would have been obvious to one of ordinary skill in the art at the time the invention was made to include the feature of having tasks of different types for the reason of making the existing system more versatile by having more variety of tasks.

Gulsen also fails to explicitly teach:

- tasks being processed in order by loading the associated code into the instruction cache for execution on the microprocessor

Kirk also teaches a processor which executes instructions for using a cache to load the tasks *(processor, cache, tasks, execution, col 17, lines 55-67)*. It would have been obvious to one of ordinary skill in the art at the time the invention was made to include the feature of processing

tasks into the cache for execution for the reason of increasing speed and productivity of the system. Memory caching is effective because most programs access the same data or instructions over and over.

Gulsen in view of Kirk fails to explicitly teach:

- placing the tasks of the same task type into a batch such that the tasks in a batch are processed before processing the next ordered task

It is notoriously common knowledge and well known in the art that a batch can contain task of the same type. In addition, Klein teaches a processing system where a batch is processed before the next task *("batch job to be presented to it before going on to perform other tasks", col. 1, lines 50-65)*. It would have been obvious to one of ordinary skill in the art at the time the invention was made to include the feature of processing a batch before other tasks to the existing method for the reason of increasing the efficiency of time *("The resultant waste of time that could otherwise be utilized if such tasks could be operated in parallel is substantial.", col. 1, lines 50-65)*.

8. Referring to claim 3, Kirk discloses the following when the task is not capable of being loaded into the cache as a whole:

- the code being logically divided at one or more break points into two or more portions *("cache partitioned evenly among 8 tasks", col 17, lines 55-57)*;
- responding to a break point defined within a first portion of the code to schedule a further task for future execution of a second portion of the code *("two tasks", "shared*

partition", "map to 1 00aa aaaa aaaa aaaa", "map to 0 0aaa aaaa aaaa aaaa) col 23 lines 64-67 and col 24, lines 1-5, and Fig. 13).

It is inherent to execute existing tasks in the future that have not been executed yet.

9. **Claims 4 and 6 are rejected under 35 U.S.C. 103(a) as being unpatentable over Gulsen (US 5,727,211) in view of Kirk (US 5,875,464), further in view of Klein (US 5,835,763) and further in view of Peters et al. (hereinafter Peters) (US 5,668,993).**

10. Referring to claim 4, Kirk discloses the following:

- a further scheduled task placed in a batch (from the rejections stated in claims 1 and 3);

Gulsen in view of Kirk, in view of Klein, and further in view of Kirk fails to explicitly teach wherein the task is placed in a batch of like tasks. However, Peters teaches that it is beneficial to group similar tasks into a batch *("To overcome this problem with multitasking, multiprocessor systems, which utilize more than one CPU, have been developed to provide tasks with the same resources offered by their uniprocessor counterparts but further allow these resources to be shared among a set of concurrently executing tasks. In multitasking, multiprocessor environments, various tasks are distributed to the various processors. A fine grain approach parallelizes groupings of similar tasks with all of the tasks being assembled into a finished batch after parallel processing completes. Coarse grain, on the other hand, simply parallelizes groupings of various tasks of the job without regard for the similarity of the tasks within each grouping.", col. 3, lines 1-13).* It would have been obvious to one of ordinary skill in the art at

the time the invention was made to include the feature of grouping similar tasks into a batch for the reason of increasing speed and efficiency through the use of parallelizing *("A fine grain approach parallelizes groupings of similar tasks with all of the tasks being assembled into a finished batch after parallel processing completes.", col. 3, lines 1-13).*

11. Referring to claim 6, Gulsen in view of Kirk, in view of Klein, and further in view of Kirk fails to explicitly teach placing a task in a batch at the same time that it is scheduled for the reason of increasing the speed of the processing. However, Peters teaches that it is beneficial in multitasking to group similar tasks into a batch. It would have been obvious to one of ordinary skill in the art at the time the invention was made to include the feature of grouping similar tasks into a batch as soon as it is scheduled for the reason of increasing speed and efficiency through the use of parallelizing *("A fine grain approach parallelizes groupings of similar tasks with all of the tasks being assembled into a finished batch after parallel processing completes.", col. 3, lines 1-13).*

12. Claim 2 is rejected under 35 U.S.C. 103(a) as being unpatentable over Gulsen (US 5,727,211) in view of Kirk (US 5,875,464), further in view of Klein (US 5,835,763) and in further view of Peters (US 6,332,167 B1).

13. Referring to claim 2, Kirk teaches the following:

- tasks being processed in order by loading the associated code into the instruction cache for execution on the microprocessor (from the rejection of claim 1 as stated above)

Gulsen in view of Kirk fails to explicitly teach:

- executing the loaded code to process the further task where there is a further task of like type in the batch

However, as stated in the rejection of claim 1, it is obvious to have a more than one similar task (or a "further task") in a batch system. From the reference of Peters, it is common knowledge that "batch processes are performed in a task oriented manner" (col 2, lines 41-42).

14. Claim 5 is rejected under 35 U.S.C. 103(a) as being obvious over Gulsen (US 5,727,211) in view of Kirk (US 5,875,464), further in view of Klein (US 5,835,763) and in further view of Nilsen (US 6,438,573 B1).

15. Referring to claim 5, Gulsen in view of Kirk fails to explicitly teach having each portion of code define an atomic operation. However, Nilsen "shows a code fragment which describes an atomic segment of code" (col 4, lines 32-33). It would have been obvious to one of ordinary skill in the art at the time the invention was made to include the feature of atomicity to the existing system of Gulsen in view of Kirk for the reason of making the system more reliable. Atomicity is beneficial because it assures that the operation follows through completely and accurately or not at all.

16. Claim 7 is rejected under 35 U.S.C. 103(a) as being obvious over Gulsen (US 5,727,211) in view of Kirk (US 5,875,464), further in view of Klein (US 5,835,763) and in further view of Servi (US 5,381,546).

17. Referring to claim 7, Gulsen in view of Kirk fails to explicitly teach having queues to manage the tasks. However, the reference of Servi illustrates that it is common knowledge that "different queues" could manage "different types of tasks" (col 2, lines 12-15) and it would have been obvious to one of ordinary skill in the art at the time the invention was made to include this feature to the existing system for the reason of having a data structure to better organize the information.

18. Claim 10 is rejected under 35 U.S.C. 103(a) as being unpatentable over Gulsen (US 5,727,211) in view of Kirk (US 5,875,464), further in view of Klein (US 5,835,763), and in further view of Bourekas (US 6,128,703).

19. Referring to claim 10, from the reference of Bourekas, it is common knowledge for a data processing apparatus to have the microprocessor and cache embodied on a single chip ("In many modern microprocessors the primary cache 208 is on the same chip with the CPU 204", col 1, lines 61-63).

20. **Claims 11-13 and 15-17 are rejected under 35 U.S.C. 103(a) as being unpatentable over Peters et al. (hereinafter Peters) (US 5,668,993) in view of Kirk (US 5,875,464).**

21. Referring to claim 11, Peters teaches a method for scheduling tasks in a task queue, the method comprising:

- identifying a new task to be scheduled in the task queue *("As batched applications comprise a plurality of tasks, and uniprocessor architectures are capable of executing only a single task at a time, uniprocessors are often complimented with special multitasking hardware and operating system software (such as UNIX) which allow the single processing resource to be efficiently distributed among a set of simultaneously initiated tasks. Although this multitasking increases a uniprocessor machine's overall throughput and workflow capabilities, the simultaneously initiated tasks are still in contention for a single processing resource and the amount of execution time allotted to each individual task decreases in proportion to the number of tasks initiated.", col. 2, lines 56-67)*;
- task queue includes a cached task *(tasks in "batch", "cache manager", "caches", col. 18, lines 10-18)*;
- batching the new task with the cached task if the task queue includes the cached task that requires substantially the same code to process the cached task and the new task *("To overcome this problem with multitasking, multiprocessor systems, which utilize more than one CPU, have been developed to provide tasks with the same resources offered by*

their uniprocessor counterparts but further allow these resources to be shared among a set of concurrently executing tasks. In multitasking, multiprocessor environments, various tasks are distributed to the various processors. A fine grain approach parallelizes groupings of similar tasks with all of the tasks being assembled into a finished batch after parallel processing completes. Coarse grain, on the other hand, simply parallelizes groupings of various tasks of the job without regard for the similarity of the tasks within each grouping.", col. 3, lines 1-13).

Peters teaches having cache, a cache manager, and tasks in a batch *(tasks in "batch", "cache manager", "caches", col. 18, lines 10-18)* but fails to explicitly teach determining if the task queue includes a cached task. However, Kirk teaches tasks using cached tasks in a task queue. It is determined that if the task queue includes a cached task, the task will be taken from the "cache partition areas" *("cache memory buffer", "cache lines allocated to a task to form a partition", "task", "cache", see Abstract).* It would have been obvious to one of ordinary skill in the art at the time the invention was made to include the feature of determining if the task queue includes a cached task to the existing method for the reason of increasing speed and efficiency by utilizing the benefits of cache *("The system cache provides a high speed access path to memory data, so that during execution of a task the logic means and registers provide any necessary cache partitioning to assure a preempted task that it's cache contents will not be destroyed by a preempting task.", col. 3, lines 5-13).*

22. Referring to claims 12 and 16, Peters teaches the method of claims 11 and 15, respectively, where batching the new task with the cached task if the task queue includes the

cached task that requires substantially the same code to process the cached task and the new task but fails to explicitly teach adding the new task to the end of the queue if the task queue does not include the cached task that requires substantially the same code to process the cached task as the new task. However, "Official Notice" is taken that both the concept and advantages of providing that items (tasks) in a queue be placed at the end (by a priority queue, for example) is well known and expected in the art. It would have been obvious to one of ordinary skill in the art at the time the invention was made to include the feature of placing unsimilar code to process the cached task as the new task toward the end of the queue to the existing method for the reason of increasing efficiency by maintaining the group of tasks with substantially the same code. As taught in the reference of Peters, "a fine grain approach parallelizes groupings of similar tasks with all of the tasks being assembled into a finished batch after parallel processing completes"*(col. 3, lines 1-13).*

23. Referring to claims 13 and 17, Kirk teaches the method of claim 11, and 15, respectively, further comprising:

- loading task code for processing the cached task into an instruction cache *("cache ID", "cache", "tasks are loaded", col. 14, lines 13-25)*;
- executing the task code for processing the cached task in the instruction cache *("The system cache provides a high speed access path to memory data, so that during execution of a task the logic means and registers provide any necessary cache partitioning to*

assure a preempted task that it's cache contents will not be destroyed by a preempting task.", see Abstract);

- executing the task code for processing the new task in the instruction cache without loading new code into the instruction cache *("The system cache provides a high speed access path to memory data, so that during execution of a task the logic means and registers provide any necessary cache partitioning to assure a preempted task that it's cache contents will not be destroyed by a preempting task. This permits use of a software controlled partitioning system which allows segments of a cache to be statically allocated on a priority I benefit basis without hardware modification to said system.", see Abstract)*.

24. Referring to claim 15, it is rejected for the same reasons as stated in the rejection of claim 11. In addition, "Official Notice" is taken that both the concept and advantages of providing that computer code stored in tangible media is well known and expected in the art. It would have been obvious to one of ordinary skill in the art at the time the invention was made to include computer code stored in tangible media to the existing method for the reason of having a portable storage medium that can be used in multiple computer systems.

25. **Claims 14 and 18 are rejected under 35 U.S.C. 103(a) as being unpatentable over Peters et al. (hereinafter Peters) (US 5,668,993) in view of Kirk (US 5,875,464) and further in view of Nilsen (US 6,438,573 B1).**

26. Referring to claims 14 and 18, Kirk teaches the method of claims 13 and 17, respectively, further comprising:

- determining if the task code is capable of fully loading into the instruction cache *("Determining the required number of segments is accomplished by execution of the task in the different legal partition sizes. The response time will decrease monotonically with increasing cache size. Once the number of cache segments required for a specific execution time is determined, the segments are dedicated to this task and removed from the cache segments available to the periodic tasks.", col. 19, lines 13-20).*

Peters in view of Kirk fails to explicitly teach :

- if the task code is not capable of fully loading into the instruction cache, logically dividing the task code such that at least one substantially atomic portion of the task code will fully load in the instruction cache.

However, Nilsen teaches the use of time slicing of task code into segments for an atomic execution *("specified code segment will execute during a plurality of time slices within an execution period, the plurality of time slices adding up to the specified increment of time, execution otherwise being aborted", col. 3, lines 47-57, "An atomic segment is executed either in its entirety or not at all", col. 12, lines 1-2, "cache", see claim 67).* It would have been obvious to one of ordinary skill in the art at the time the invention was made to include the feature of atomic time slicing if the task code cannot be fully loaded to the existing method for the reason of being able to load the task code no matter the size and to do so accurately.

ARGUMENTS

27. Applicant's arguments with respect to claims 1-10 have been considered but are moot in view of the new ground(s) of rejection.

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Kenneth Tang whose telephone number is (703) 305-5334. The examiner can normally be reached on 9:00am-6:30pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, William Grant can be reached on (703) 308-1108. The fax phone number for the organization where this application or proceeding is assigned is (703) 872-9306.

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the receptionist whose telephone number is (703) 746-7140.

Kt
September 2, 2003

MAJID A. BANANKHAH
PRIMARY EXAMINER